A New Approach for Inversion of Large Random Matrices in Massive MIMO Systems

Muhammad Ali Raza Anjum*, Muhammad Mansoor Ahmed

Department of Electronic Engineering, Mohammad Ali Jinnah University, Islamabad, Pakistan

Abstract

We report a novel approach for inversion of large random matrices in massive Multiple-Input Multiple Output (MIMO) systems. It is based on the concept of inverse vectors in which an inverse vector is defined for each column of the principal matrix. Such an inverse vector has to satisfy two constraints. Firstly, it has to be in the null-space of all the remaining columns. We call it the null-space problem. Secondly, it has to form a projection of value equal to one in the direction of selected column. We term it as the normalization problem. The process essentially decomposes the inversion problem and distributes it over columns. Each column can be thought of as a node in the network or a particle in a swarm seeking its own solution, the inverse vector, which lightens the computational load on it. Another benefit of this approach is its applicability to all three cases pertaining to a linear system: the fully-determined, the over-determined, and the under-determined case. It eliminates the need of forming the generalized inverse for the last two cases by providing a new way to solve the least squares problem and the Moore and Penrose's pseudoinverse problem. The approach makes no assumption regarding the size, structure or sparsity of the matrix. This makes it fully applicable to much in vogue large random matrices arising in massive MIMO systems. Also, the null-space problem opens the door for a plethora of methods available in literature for null-space computation to enter the realm of matrix inversion. There is even a flexibility of finding an exact or approximate inverse depending on the null-space method employed. We employ the Householder's null-space method for exact solution and present a complete exposition of the new approach. A detailed comparison with well-established matrix inversion methods in literature is also given.

Citation: Anjum MAR, Ahmed MM (2014) A New Approach for Inversion of Large Random Matrices in Massive MIMO Systems. PLoS ONE 9(4): e94958. doi:10.1371/journal.pone.0094958

Editor: Shyamal D Peddada, National Institute of Environmental and Health Sciences, United States of America

Received November 26, 2013; **Accepted** March 21, 2014; **Published** April 14, 2014

Funding: The authors have no support or funding to report.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: ali.raza.anjum@gmail.com

Introduction

Multiple-Input Multiple-Output (MIMO) systems form a well established area of wireless communications [1]. A significant increase in interest in MIMO systems has been witnessed in the last few years due to the advent of massive MIMO systems [2,3]. In these systems, more degrees of freedom in terms of data rate and link reliability are available due to the increase in the number of transmit and receive antennas. These advantages become even more impressive in multi-user scenario when there is a possibility to transmit to several users simultaneously [2]. However, there is still a challenge of low complexity detection techniques for the practical realization of such systems [4]. Many high performance detection methods for massive MIMO systems require an unconstrained solution to a linear estimation problem [5]. Linear estimation requires the inversion of channel matrix which, in such systems, can be problematic because of its potentially large size. For example, matrices of size 40*40 and above have recently been reported in literature as massive [2].

Therefore, there is a need to find solutions that do not require outright inversion. Various methods in this regard have been proposed in literature: Cayley-Hamilton method, Neumann series method, QR method, random matrix methods, LSMR, LSQR, Kaczmarz method and the ones based on polynomial and truncated polynomial filters [6–19]. These methods still require a lot of computational effort and some of them have even proven to be more complex than the outright inversion [2,6]. While it is possible that some may perform better than other, it is not always possible to have a fair comparison among them since these methods follow independent approaches and are dependent on different sets of parameters.

Keeping that in view, our objective in this paper is to develop an alternate method to find the inverse of the channel matrix. While addressing this problem, we dispense with three fundamental assumptions that are generally implied in the traditional methods. We endeavor to do so in order to provide more room for thought. First, no assumption has been made about the structure of the channel matrix. Secondly, the matrix can be purely random instead of being deterministic. Finally, we do not assume that it is sparse [20]. Keeping this in view, we report a novel method that not only finds the inverse of a channel matrix but also brings a new perspective to the inversion process itself. The proposed method is a comprehensive one and is fully applicable to all three matrix inversion cases pertaining to a linear system. We also provide its detailed comparison with well-known matrix inversion methods available in literature for the sake of analysis and, hence, understanding.

Rest of the article is organized as follows. To begin with, the subsequent section lays out basic system model and the essential nomenclature. The third section presents the basic idea behind the proposed method. The inversion problem is solved according to the proposed method using Householder's null-space method in the fourth section. In fifth section, solution to three fundamental cases of linear systems is presented. Proposed method is then compared with the QR decomposition (QRD) method, Least Squares (LS) method, and Moore and Penrose's pseudoinverse method in sections six, seven, and eight respectively. A complete algorithm for step by step computation of the inverse matrix according to the proposed method is outlined in section nine. Simulation results demonstrating the speed and accuracy of the proposed method in comparison to the state of the art methods available in literature are presented in section ten. Finally, the article concludes with a brief discussion in section eleven.

System Model

A MIMO system is represented by a system of linear equations [21].

$$Hx=y \tag{1}$$

H is the channel matrix of dimension $m \times n$, x is the transmitted vector of dimension $n \times 1$, and y is the received vector of dimension $m \times 1$. Entries of H matrix are complex-valued independent and identically distributed Gaussian random variables of zero mean and unity variance [6]. m and n are the number of transmit and receive antennas respectively. If $m=n$, the number of equations is equal to the number of unknowns and the system is fully-determined. If $m>n$, the number of equations is greater than the number of unknowns and the system is over-determined. If $m<n$, the number of equations is less than the number of unknowns and the system is under-determined.

Basic Idea

We begin with the assumption that H has a full-rank. Re-writing Eq. (1) as:

$$y=h_j x_j+\sum_{\substack{i=1\\ i\neq j}}^{n} h_i x_i \tag{2}$$

First term on the right hand side of Eq. (2) denotes j-th column of matrix $H.x_j$ is the associated unknown. The second term represents the remaining columns of H with their respective unknowns. This term can be separately written in matrix form by defining a new column reduced matrix F_j and a symbol reduced vector $\tilde{x}$.

$$y=h_j x_j+F_j\tilde{x} \tag{3}$$

F_j has dimensions of $m \times (n-1)$. Taking the dot product of Eq. (3) with an arbitrary column vector w_j of dimension $m \times 1$,

$$w_j^H y=w_j^H h_j x_j+w_j^H F_j\tilde{x} \tag{4}$$

To evaluate the j-th unknown x_j from Eq. (4),

$$w_j^H h_j=1 \tag{5}$$

$$w_j^H F_j=0 \tag{6}$$

Eqs. (5) and (6) define the j-th row of the inverse matrix. It should have a dot product of one with the j-th column and zero with all the remaining columns. The same is also true for all the other rows of the inverse matrix. Once we are able to solve Eqs. (5) and (6) for w_j, all rows of inverse matrix can be computed one by one in the same fashion and a complete inverse matrix can be built. Eq. (5) essentially restricts the length of the null-space solution produced by Eq. (6) and serves as a constraint for Eq. (6). Therefore, we proceed by solving the Eq. (6) first. It can be solved by a variety of methods available in literature of for computing the null-space of a matrix. We propose to solve it by Householder's null-space method because it is stable and provides an exact solution.

Solution Using Householder's Null-Space Method

Householder's method has been traditionally viewed as a means to achieve the QR decomposition (QRD). It performs a series of orthogonal transformations on an arbitrary matrix to convert it to an upper triangular matrix. These transformations can be performed either by Given's rotation matrices or Householder's reflection matrices. A reflection matrix can perform the job of many rotation matrices at once. Therefore, reflection matrices are more desirable in our case. We proceed by performing a series of orthogonal transformations on the matrix F_j using the reflection matrices to convert it to an upper triangular matrix R. $n-1$ reflection matrices will be required for that purpose because F_j has $n-1$ columns.

$$H_{n-1}\dots H_3 H_2 H_1 F_j=QF_j=R \tag{7}$$

H_i's represent the Householder's reflection matrices. They are square, symmetric and orthogonal with each one having the dimensions of $m \times m$. Q has also the same dimensions. But the dimensions of R are $m \times (n-1)$. This is because the first $n-1$ rows of Q generate an $(n-1)\times(n-1)$ upper triangular portion in R. Remaining $m-(n-1)$ rows in R are zero. These rows are produced by last $m-(n-1)$ rows of Q.

$$R=[\ r_1 \quad r_2 \quad \dots \quad r_{n-1} \quad 0 \quad \dots \quad 0 \quad 0\ ]^H$$

$$Q=[\ q_1 \quad q_2 \quad \dots \quad q_{n-1} \quad \hat{q}_n \quad \dots \quad \hat{q}_{m-1} \quad \hat{q}_m\ ]^H \tag{8}$$

$$R=[\ r_1 \quad r_2 \quad \dots \quad r_{n-1} \quad 0 \quad \dots \quad 0 \quad 0\ ]^H \tag{9}$$

$\hat{q}$'s mark the rows that produce zeros in R matrix. Substituting Eqs. (8) and (9) in Eq. (7),

$$\begin{bmatrix} q_1^H \\ q_2^H \\ \vdots \\ q_{n-1}^H \\ \hat{q}_n^H \\ \vdots \\ \hat{q}_{m-1}^H \\ \hat{q}_m^H \end{bmatrix} F_j = \begin{bmatrix} r_1^H \\ r_2^H \\ \vdots \\ r_{n-1}^H \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \tag{10}$$

Concentrating on $\hat{q}$'s only in Eq. (10),

$$\begin{bmatrix} \hat{q}_n^H \\ \vdots \\ \hat{q}_{m-1}^H \\ \hat{q}_m^H \end{bmatrix} F_j = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{11}$$

$\hat{q}$'s form the basis for the left null-space of F_j and, hence, solve Eq (6). Only equation that remains to be solved now is Eq. (5). In order to do so, we would need to consider the three fundamental cases of linear systems.

Three Fundamental Cases

In this section, we take up the three fundamental cases of linear systems and solve them one by one.

1. The Fully-Determined Case

In the fully-determined case, there is only one vector in the left null-space of F_j, i.e., $m-(n-1)=1$ given that $m=n$. The solution in Eq. (11) is unique.

$$\hat{q}_j F_j = 0 \tag{12}$$

The only remaining task is to rescale $\hat{q}_j$ according to Eq. (5).

$$w_j^H = \frac{\hat{q}_j^H}{k_j} \tag{13}$$

Whereas,

$$k_j = \hat{q}_j^H h_j \tag{14}$$

k_j is the rescaling factor for the j-th row of inverse matrix F_j. By iterating j for all columns of H matrix, complete inverse matrix W can be built.

$$W = \begin{bmatrix} w_1^H \\ w_2^H \\ \vdots \\ w_n^H \end{bmatrix} = \begin{bmatrix} \frac{\hat{q}_1^H}{k_1} \\ \frac{\hat{q}_2^H}{k_2} \\ \vdots \\ \frac{\hat{q}_n^H}{k_n} \end{bmatrix} \tag{15}$$

such that,

$$WH = \begin{bmatrix} w_1^H \\ w_2^H \\ \vdots \\ w_n^H \end{bmatrix} [h_1 \quad h_2 \quad \dots h_n] = \begin{bmatrix} \frac{\hat{q}_1^H}{k_1} \\ \frac{\hat{q}_2^H}{k_2} \\ \vdots \\ \frac{\hat{q}_n^H}{k_n} \end{bmatrix} [h_1 \quad h_2 \quad \dots h_n]$$

$$= \begin{bmatrix} \frac{\hat{q}_1^H h_1}{k_1} & \frac{\hat{q}_1^H h_2}{k_1} & \cdots & \frac{\hat{q}_1^H h_n}{k_1} \\ \frac{\hat{q}_2^H h_1}{k_2} & \frac{\hat{q}_2^H h_2}{k_2} & \cdots & \frac{\hat{q}_2^H h_n}{k_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\hat{q}_n^H h_1}{k_n} & \frac{\hat{q}_n^H h_2}{k_n} & \cdots & \frac{\hat{q}_n^H h_n}{k_n} \end{bmatrix}$$

$$WH = \begin{bmatrix} \frac{k_1}{k_1} & & & \\ & \frac{k_2}{k_2} & & \\ & & \ddots & \\ & & & \frac{k_n}{k_n} \end{bmatrix} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix}$$

The identity matrix will have the dimensions of $\mathrm{n} \times \mathrm{n}$.

2. The Over-Determined Case

In the over-determined case, the matrix H will already have a right null-space due to a greater number of rows than the columns. The removal of extra column to form F_j will populate the null-space even further and there will be $m-(n-1)$ vectors in the left null-space of F_j. Hence, the solution to the Eq. (6) will not be unique.

$$\begin{bmatrix} \hat{q}_n^H \\ \vdots \\ \hat{q}_{m-1}^H \\ \hat{q}_m^H \end{bmatrix} F_j = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{16}$$

All the vectors from $\hat{q}_n$ to $\hat{q}_m$ solve Eq. (6). An obvious question would be which one to choose. At this point, we adjourn this question to Section VIII where it would be amply addressed. For now, any one of them can be selected. The next step would be to rescale it according to Eq. (13). By repeating the same procedure for all columns of H, a complete inverse matrix can be built.

3. The Under-Determined Case

In the under-determined case, H matrix will have right null-space due to a greater number of columns than the rows. In order to be consistent with the earlier work, we can transpose it and move the null-space to left. The inverse matrix can then be computed in the manner described in subsection (b). Only a transposition of inverse matrix is required afterwards to revert to the original case.

Comparison with QR Decomposition

Traditional methods employing QR decomposition take a linear system of the form,

$$Ax = b \tag{17}$$

into,

$$x = R^{-1} Q^{H} b \tag{18}$$

by factorizing A in Eq. (17) into an orthogonal matrix Q and an upper-triangular matrix R [22]. There are two famous ways for achieving QR decomposition: Gram-Schmidt QR and Householder's QR [22]. Traditionally, Gram-Schmidt method has been used for that purpose. Objective was to convert an $m \times n$ matrix A into an $m \times n$ orthogonal matrix Q. Matrix R connects A to Q.

$$A_{mn} = Q_{mn} R_{nn} \tag{19}$$

Householder's method takes a shift in perspective. R is targeted instead of Q. Being an upper triangular matrix, R is much more suitable for back substitution. Q serves as the connecting matrix.

$$A_{mn} = Q_{mm} R_{mn} \tag{20}$$

The Q in Eq. (20) has extra $m-n$ columns than the one obtained by Gram-Schmidt method. First m columns serve as the orthonormal basis for the column space of A. Remaining $m-n$ columns form the basis for the left null-space of A because R can only be made triangular upto first n rows. [22].

$$A_{mn} = \begin{bmatrix} Q_{mn} & Q_{m(m-n)} \end{bmatrix} \begin{bmatrix} R_{nn} \\ 0_{(m-n)n} \end{bmatrix} \tag{21}$$

In the proposed method, Househoder's method is employed to find the null space basis $Q_{m(m-n)}$ of F_j. Also, there is a difference in the orientation of Q in Eq. (8) which leads to,

$$A = Q^{H} R \tag{22}$$

instead of $A = QR$. We take Q as such because its last $m-n$ rows are responsible for the zero last $m-n$ rows of R. Orthonormal column space of Q in Eq. (20) is now an orthonormal row space in Eq. (22).

Comparison with Least Squares

When A has fewer columns than the rows, the system becomes over-determined. There are more equations than the unknowns. The solution is then obtained by Least Squares (LS).

$$A^{H} A x = A^{H} b \tag{23}$$

QR decomposition is a popular method to solve LS problems and when applied to Eq. (23) will result in Eq. (18).

We have not approached the over-determined case using LS. We have taken it column by column. A dearth of columns in an over-determined matrix signals an apriori presence of the left null-space. Removal of one more column will populate it further. This will bring us to select, out of this abundance of null-space vectors, the one which has the highest value of k depicted in Eq. (14). It needs to be so because the selected vector will then be divided by k according to Eq. (13). Smaller values of k can inflate the magnitude of the null-space vector after division. In the worst case of $k = 0$, there will be a division by zero. Therefore, out of many null-space vectors obtained by Householder's null-space method, the best would be the one which has the highest value of k for the specific column vector. The term best can be pushed even further. In ideal case, a value of one for k would imply no division at all. But the caveat is to avoid smaller values of k in order to prevent instability.

Comparison with Moore and Penrose's Pseudoinverse

Now we turn to the last case, the case when the matrix A is singular. A solution in this case is sought by forming the Moore and Penrose's pseudoinverse. Not an exact inverse though, the idea is to find the shortest vector that solves the LS equation. The term shortest implies a vector that has no null-space component. One way to do is to use Singular Value Decomposition (SVD) [20]. SVD decomposes the matrix A into two square, orthonormal matrices U and V. These matrices contain the basis for the column and row space of A respectively.

$$A = U \sum V^{H} \tag{24}$$

The third matrix $\sum$ is a diagonal matrix that contains the singular values. Some of the singular values in $\sum$ will be zero for this case due to the presence of dependent columns in A. Columns of U and V for the corresponding zero singular values will represent the left and right null-spaces of A. Pseudoinverse is formed by leaving the zero singular values unchanged while inverting the others. This gives the shortest solution in row space.

At this point, a 2×2 matrix with all entries equal to one would serve best to explain our approach. Once the first column is removed, Eq. (6) will seek a solution in the left null-space of the remaining columns. Eq. (5) will dictate this solution to have a projection of value equal to one with the removed column. This is impossible because both the removed column and the remaining column are exactly the same. A vector cannot be orthogonal and parallel to itself at the same time. In our opinion, this is precisely the reason for non-invertibility of a matrix rather than the traditional zero-determinant view. The only unique vector is the

Table 1. Comparison of the proposed algorithm with the state of the art methods available in literature.

M = N	Proposed	LSMR			LSQR			Kaczmarz 100 iterations		
	time	||r||	||e||	time	||r||	||e||	time	||r||	||e||	time
N = 20	0	1.6547	17.9982	0.0625	0.0511	0.0167	0.0781	2.6355	21.7263	0.0469
N = 40	0.0156	0.0795	0.2899	0.0313	0.6181	2.3153	0.0938	0.4677	1.2842	0.1406
N = 60	0.0313	1.2373	7.7615	0.0469	2.4645	11.5742	0.0625	2.9948	10.7212	0.2969
N = 80	0.0625	0.4229	2.4126	0.0469	1.0575	4.8856	0.0625	1.3186	5.6604	0.3906
N = 100	0.1250	0.6446	5.5752	0.0625	2.3165	9.5884	0.0938	2.5353	8.4609	0.5156
N = 120	0.2500	1.2121	38.2089	0.0469	3.0198	39.6965	0.0625	3.2040	39.9166	0.7656
N = 140	0.4375	1.4229	13.3729	0.0781	3.0258	18.4997	0.0625	4.1424	18.8697	1.0625
N = 160	0.7344	0.4929	8.3269	0.0469	1.7582	9.7641	0.0781	1.5562	9.8529	1.3281
N = 180	1.2031	0.8819	7.1348	0.0625	2.9170	13.5627	0.0781	3.0961	13.4074	1.7031
N = 200	1.7656	0.9897	19.4026	0.0625	3.0598	24.0314	0.0781	4.2552	24.1649	2.0781

doi:10.1371/journal.pone.0094958.t001

vector left in the column space of F, the only remaining column. Therefore, it is rescaled with itself to satisfy Eq. (13). This is exactly the solution obtained by applying the SVD to above problem.

Algorithm

We summarize the steps discussed in the form of a complete algorithm as follows.

1. If $m \geq n$, continue to step 2, otherwise transpose the matrix H before moving to step 2.
2. Locate the j-th column whose inverse vector is to be determined.
3. Remove that column from H matrix to form a new matrix F_j.
4. Decompose F_j to form Q and R matrices according to Eq. (7).
5. Discard the R matrix and the first n rows of Q matrix.
6. Remaining $m-n$ rows will constitute null-space of F_j. In case $m=n$, there will be only one row and, hence, one candidate for the inverse channel vector. In case $m>n$, all of the remaining $m-n$ rows satisfy the criteria in Eq. (6). In this case, pick any one at random.
7. Normalize the selected vector according to Eqs. (13) and (14) to form the inverse vector w_j^H.
8. Go back to step 2 and repeat the same procedure for the remaining columns of H matrix.
9. Stack the inverse channel vectors on top of each other in the manner described in Eq. (15) to form the complete inverse matrix W.
10. For the case when $m<n$, transposition of W formed in step 9 is necessary to form the right handed inverse.

A user-friendly code of this algorithm is provided in Code S1.

Simulation Results

We now present the simulation results of the algorithm. The components of the channel matrix are chosen to be independent and identically distributed (IID) circularly symmetric Gaussian random variables with a zero mean and unity variance. m is selected to be equal to n as this refers to multi-user case in M-MIMO systems because both the number of transmitting and receiving antennas become very large. For example, a typical matrix size can be 40×40 and above [2]. Also when $m=n$, an exact solution is possible and the residue and hence the error in the estimate can be zero.

For simulation purpose, various matrix sizes have been selected and the results obtained in terms of the norm of residue, norm of the error in estimate and the computational time taken are displayed in Table 1 against the state of the art algorithms available in literature: LSMR, LSQR, and Kaczmarz method. Since the proposed method achieves zero error/residue norms, its respective columns are not displayed in the table. Codes required for simulation of LSQR and LSMR algorithms have been downloaded from the website of Stanford University's System Optimization Laboratory and are used as is.

Simulation results demonstrate that when LSMR, LSQR, and Kaczmarz method are applied to this problem, these methods suffer from very large residue norms. But that is not all. The norms for the error in estimate are even worse. Kaczmarz method is not only the slowest of all but also has largest error/residue norms. Hence, the estimates become practically useless. On the other hand, the proposed method achieves perfectly zero error/residue norms in moderate time. Therefore, the proposed method can be a much better choice in this context due to its better accuracy and speed.

Conclusion

A novel approach for matrix inversion has been presented in this paper. The matrix was split into a set of column reduced matrices, each with its own inverse channel vector w_j. In order to qualify for an inverse channel vector, w_j had to satisfy two constraints; $w_j^H F_j = 0$ and $w_j^H h_j = 1$. Householder's null-space method was used to solve the first constraint and scaling was used to solve the second one. Inverse matrix was then built from these inverse channel vectors. A detailed comparison with the state of the art methods available in literature was carried out all along to emphasize the novelty of the method.

Supporting Information

Code S1 A user-friendly code of the proposed algorithm.
(ZIP)

Author Contributions

Conceived and designed the experiments: MARA. Performed the experiments: MARA. Analyzed the data: MARA. Contributed reagents/materials/analysis tools: MARA. Wrote the paper: MARA. Played the role of supervisor in this research: MMA.

References

1. Paulraj AJ, Gore DA, Nabar RU and Bolcskei H (2004) An overview of MIMO communications - a key to gigabit wireless. Proceedings of the IEEE 92: 198–218.
2. Rusek F, Persson D, Buon Kiong L, Larsson EG, Marzetta TL, et al. (2013) Scaling Up MIMO: Opportunities and Challenges with Very Large Arrays. IEEE Signal Processing Magazine 30: 40–60.
3. Vishnu Vardhan K, Mohammed SK, Chockalingam A and Sundar Rajan B (2008) A low-complexity detector for large MIMO systems and multicarrier CDMA systems. IEEE Journal on Selected Areas in Communications 26: 473–485.
4. Yi W and Leib H (2013) Sphere Decoding for MIMO Systems with Newton Iterative Matrix Inversion. IEEE Communications Letters 17: 389–392.
5. Hochwald BM and Ten Brink S (2003) Achieving near-capacity on a multiple-antenna channel. IEEE Transactions on Communications 51: 389–399.
6. Couillet R and Debbah M (2011) Random Matrix Methods for Wireless Communications. Cambridge University Press. 568 p.
7. Edman F and Owall V (2003) Implementation of a scalable matrix inversion architecture for triangular matrices. 14th IEEE 2003 International Symposium on Personal, Indoor and Mobile Radio Communications Proceedings, 7-10 Sept 2003. Piscataway, NJ, USA: IEEE. pp. 2558–2562.
8. Golub GH and Loan CFV (1996) Matrix computations (3rd ed.). Johns Hopkins University Press. 694 p.
9. Honig ML and Weimin X (2001) Performance of reduced-rank linear interference suppression. IEEE Transactions on Information Theory 47: 1928–1946.
10. Karkooti M, Cavallaro JR and Dick C (2005) FPGA implementation of matrix inversion using QRD-RLS algorithm. 2005 39th Asilomar Conference on Signals, Systems and Computer, 30 Oct-2 Nov 2005. Piscataway, NJ, USA: IEEE. pp. 1625–1629.
11. Lei M, Dickson K, McAllister J and McCanny J (2011) QR Decomposition-Based Matrix Inversion for High Performance Embedded MIMO Receivers. IEEE Transactions on Signal Processing 59: 1858–1867.
12. Muller RR and Verdu S (2001) Design and analysis of low-complexity interference mitigation on vector channels. IEEE Journal on Selected Areas in Communications 19: 1429–1441.
13. Stewart GW (2001) Matrix algorithms. Society for Industrial and Applied Mathematics. 469 p.
14. Wu D, Eilert J, Liu D, Wang D, Al-Dhahir N, et al. (2007) Fast complex valued matrix inversion for multi-user STBC-MIMO decoding. IEEE Computer Society Annual Symposium on VLSI: Emerging VLSI Technologies and Architectures, ISVLSI'07, March 9, 2007 - March 11, 2007. Porto Alegre, Brazil: Inst. of Elec. and Elec. Eng. Computer Society. pp. 325–330.
15. S . Moshavi, E. G . Kanterakis and Schilling DL (1996) Multistage linear receivers for DS-CDMA systems. International Journal of Wireless Information Networks.
16. J. M . Chaufray, W . Hachem and Loubaton P (2004) Asymptotic analysis of optimum and sub-optimum CDMA downlink MMSE receivers. IEEE Transactions on Information Theory: 2620–2638.
17. Chong EKP and Zak SH (2013) An Introduction to Optimization. Wiley.
18. Fong DC-L and Saunders MA (2011) LSMR: An iterative algorithm for sparse least-squares problems. SIAM Journal on Scientific Computing.
19. Paige CC and Saunders MA (1982) LSQR: An Algorithm for Sparse Linear Equations and Sparse Least Squares. ACM Trans Math Softw 8: 43–71.
20. Strang G (2007) Computational Science and Engineering. {Wellesley-Cambridge Press}.
21. Tse D and Viswanath P (2005) Fundamentals of wireless communication. Cambridge University Press. 554 p.
22. Strang G (1986) Introduction to Applied Mathematics. Wellesley-Cambridge Press.